Invention Journal of Research Technology in Engineering & Management (IJRTEM)
ISSN: 2455-3689
www.ijrtem.com Volume 2 Issue 5 ǁ May 2018 ǁ PP 66-73

Clustering based Time Slot Assignment Protocol for Improving Performance in Underwater Acoustic Networks

Muzaffar Ergashev[1] and Sunmyeng Kim[2]
[1,2,] *Department of Software engineering, Kumoh National Institute of Technology, Gumi 730-701, South Korea*

ABSTRACT : *Recently, numerous approaches have been proposed for designing medium access control (MAC) in underwater acoustic networks (UANs). Some of those works tried to adapt MAC protocols proposed for terrestrial networks. However, unique environmental characteristics of UANs make the MAC protocols hard to be used in the UANs and degrade network performance. In order to improve network performance, COD-TS MAC protocol was proposed. COD-TS focuses on both single hop and multi-hop mode and utilizes CDMA for exchanging schedule information between cluster heads. COD-TS has shortcomings such as collisions, additional energy consumption by exchanging schedule information and near-far effect of CDMA. To overcome above shortcomings, we propose a clustering-based time slot assignment protocol. In the proposed protocol, nodes are clustered, and each cluster head performs two-hop neighbor cluster discovery operation. And then, a cluster head obtains its own relative position information. Finally, the cluster head assigns its own time slot for data transmission based on the information. Simulation results show that the proposed protocol has always better performance compared to the COD-TS.*

KEYWORDS: *clustering, MAC, time slot, underwater acoustic networks*

I. INTRODUCTION

Underwater acoustic networks (UAN) has become a popular topic among both academic and industrial researchers. Because UANs are the very effective system for exploring and observing under oceans. UAN is the system which consists of sensor nodes, autonomous underwater vehicles (AUV) and floating buoys which utilizes acoustic waveforms for communication among each other and all together do the special task. Main utilization purposes of UANs are ocean sampling, environmental monitoring, undersea explorations, disaster prevention, assisted navigation, distributed tactical surveillance, mine reconnaissance [1]. In early times numerous proposal of designing MAC for UANs had been done by researchers which are based on contention or handshaking algorithms. The propagation speed of acoustic (1500 m/s in water) wave is very slow comparing to the propagation speed of the electromagnetic wave (3 x 10^8 m/s), long preamble (1.5 seconds) and low transmission speed (667 bps) [2] degrades the efficiency of above mentioned protocols. Low propagation speed, the long preamble, and low transmission speed cause the high probability of simultaneous transmission of frames from different nodes in random access and handshaking protocols. In turn, it brings low throughput and low energy efficiency in communication. So, directly implementation or adjusting of existing MAC protocols of terrestrial networks for UANs will not give an effective result.

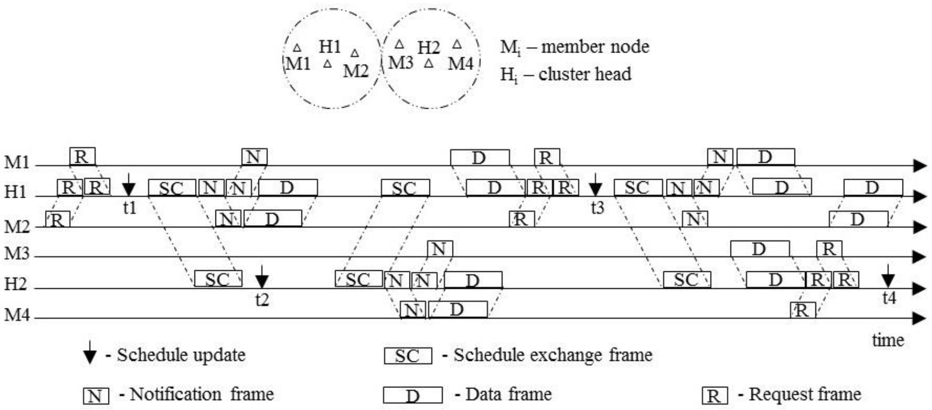


Fig. 1. An example of COD-TS with the given topology.

Several MAC protocols have been proposed for UANs and some of them are given here. In [3], communication protocol proposed for underwater vehicles. Authors proposed closed loop distributed algorithm for setting optimal transmit power and code length in CDMA [4]. Protocol proposed in [5], uses RTS/CTS packets to reserve channel, but there is still collision between data-control packets. The work in [6], uses the selection of packet size for communication in multi hop networks. In [7], improved CSMA (Distance Aware CSMA) protocol is proposed. In [8], multiband modems are used and if noise is detected in currently utilizing band, modem switches to another frequency band. Protocol designed for centralized networks in [9]. Nodes are mapped to the network schedule on its local clock according to the relative propagation delay. Those protocols mentioned above still have shortcomings and cannot provide high performance for large size and dense network.

To improve the performance of UANs COD-TS has been proposed in [2]. COD-TS uses algorithms presented in [10, 11] for synchronizing nodes. And organizes nodes into clusters using algorithm proposed in [12]. A cluster consists of a cluster head node and one-hop neighbor nodes of the cluster head. Each cluster head schedules data/control packet transmissions for the duration of time which is called communication round. Communication round is limited with D max and would be between two successive schedule updates of the cluster head. In Fig. 1, for example, the period between time t1 and t3 is one communication round of cluster head H1's cluster. In Fig. 1 each cluster has cluster heads H1 and H2, and member nodes M1 and M2, and M3 and M4, respectively. In the communication round which ends at t1, the cluster head H1 receives the request packets from member nodes M1 and M2. At time t1, the cluster head H1 makes a new schedule considering the request packets and sends it in the schedule and notification packets. The member nodes transmit data/request packets at its own time slots. The cluster head H2 receives the schedule packet from the cluster head H1 at time t2, it makes a new schedule based on the schedule packet and request packets from its member nodes. In COD-TS schedule packets are transmitted at high power and it causes to increase the interference and collision probability at between transmissions of neighboring nodes/clusters. Also, additional energy is consumed for transmissions of schedule packets which accelerates depletion of battery energy. Power control algorithm is not provided to overcome near-far effect in CDMA.

To overcome above shortcomings, we propose Clustering based Time Slot Assignment Protocol – CTSA protocol. In the proposed protocol, sensor nodes are organized into clusters, and each cluster head performs two-hop neighbor cluster discovery operation. So, each cluster head knows its two-hop neighbor clusters ids. And then, using two-hop neighbor clusters ids a cluster head obtains its own relative position information according to its relative position. Finally, the cluster head assigns its own time slot for data transmission based on its own relative position information. All clusters own their slot time and the cluster head schedules data/control packet transmissions of cluster members. In this way, collisions of transmission of neighboring nodes in the same cluster and neighboring clusters are prevented, and it is done without exchanging schedule information. This paper is organized as follows: In Section 2, we explain the proposed protocol in detail; In Section 3, we analyze network performance of the proposed protocol through the simulation and compare with COD-TS protocol; Finally, Section 4 concludes the paper.

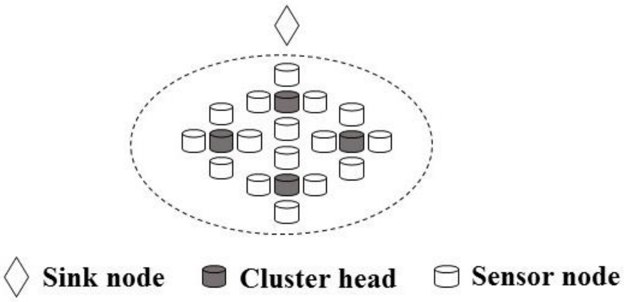


Fig. 2. An example topology of UANs.

II. PROPOSED CTSA PROTOCOL

We assume the network is static or could be seen as a static for relatively long enough duration. The network consists of a sink node and sensor nodes which is shown as an example topology in Fig. 2. We have to mention that CTSA protocol can work for the network with randomly located nodes. Also, we assume that network is well synchronized which is obtainable utilizing algorithms in [10, 11]. Time is divided into cluster times and each cluster time consists 4 intra cluster time as shown in Fig. 6. Using our algorithm, a cluster head obtains an intra

cluster time according to its relative position to its neighbor clusters. Our algorithm contains 3 phases: initialization, obtaining relative position information, and obtaining intra cluster time. Initialization phase includes clustering nodes after deployment and two hop neighbor discovery. In obtaining relative position information phase, cluster heads are assigned parameters according to their relative position to their neighbor clusters. In obtaining intra cluster time phase, each cluster head obtains a portion of time called intra cluster time for data/control packet transmissions of the cluster members according to their relative position information.

Initialization phase: In the initialization phase, first, nodes are clustered. Each cluster consists of a cluster head and member nodes

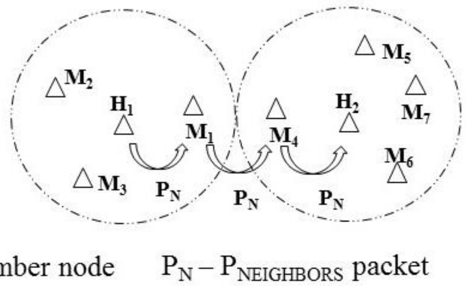
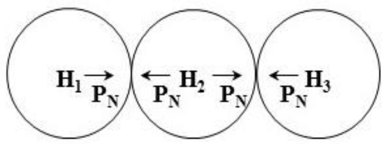


Fig. 3. Exchanging PNEIGHBORS packet.

which are one-hop neighbors of the cluster head. We use existing algorithm proposed in [12] for clustering. Second, each cluster head performs two-hop neighbor cluster discovery operation. A cluster head owns a neighbor clusters table. Table entries are the number of hops between neighbors and ids of neighbors. Each cluster head fills its table by exchanging $P_{NEIGHBORS}$ packets as shown in Fig. 3. $P_{NEIGHBORS}$ packet contains source id, N-number of one hop neighbor clusters and their ids. If a cluster head does not receive $P_{NEIGHBORS}$ packet during a fixed τ_{max} time, neighbor cluster discovery step will end. Neighbor discovery step is performed periodically at fixed ξmax time. An example of neighbor discovery step is shown in Fig. 3 and Fig. 4. The Fig. 3 shows an example of $P_{NEIGHBORS}$ packet exchange between cluster heads H1 and H2 via their members M1 and M4. $P_{NEIGHBORS}$ packet sent by cluster head H1 travels through H1 → M1 → M4 → H2 to cluster head H2. And H2 cluster head also sends $P_{NEIGHBORS}$ packet which goes through H2 → M4 → M1 → H1 to cluster head H1. An example of neighbor cluster discovery with details of updating neighbor cluster table is given in Fig. 4. Cluster head H2 sends packet P1. Since its neighbor cluster table is empty in step 0, the cluster head H2 includes only its id in source field $P_{NEIGHBORS}$ packet and sets the number of neighbor clusters field of the packet to zero. After receiving packet P1 cluster heads H1 and H3 inserts id of cluster head H2 to their table as one hop neighbor cluster as shown in step 1. Cluster head H1 sends packet P2,



Step 0

H_1's table

N_{Hop}	id0	id1	…	idn

H_2's table

N_{Hop}	id0	id1	…	idn

H_3's table

N_{Hop}	id0	id1	…	idn

Step 1

H_1's table

N_{Hop}	id0	id1	…	idn
1	H_2			

H_2's table

N_{Hop}	id0	id1	…	idn
1	H_1	H_3		

H_3's table

N_{Hop}	id0	id1	…	idn
1	H_2			

Step 2

H_1's table

N_{Hop}	id0	id1	…	idn
1	H_2			
2	H_3			

H_2's table

N_{Hop}	id0	id1	…	idn
1	H_1	H_2		

H_3's table

N_{Hop}	id0	id1	…	idn
1	H_2			
2	H_1			

P_N $P_{NEIGHBORS}$ packet **H_i i^{th} cluster's id**

Fig. 4. An example of neighbor cluster discovery.

which includes its id and one hop neighbor cluster H2's id. The cluster head H2 inserts cluster head H1's id to its table as one hop neighbor cluster as shown in step 1. Next, cluster head H3 sends packet P3 with self id and cluster head H2's id. Cluster head H2 inserts H3's to its table as shown in step 1. Since cluster head H2 updated its table it sends packet P4. Packet P4 includes H2's self id and its one hop neighbor clusters' ids H1 and H3. Cluster heads H1 and H3 receive packet P4 and H1 inserts H3's id as two hop neighbor cluster and H3 also inserts H1's id as two hop neighbor cluster to their respective tables as shown in step 2.

Obtaining relative position information: In this subsection, we explain obtaining relative position information which are a number of hops and an index number (H, I). Fig. 2 shows that the sensor network includes a sink node and sensor nodes. As shown in Fig. 5, sensor nodes are organized into clusters. CH_n is the identification number of the cluster. The sink node assigns the closest cluster head to itself as a super cluster head. In Fig. 5, super cluster head's identification number is CH_{00}. H is the number of hop between each cluster and the super cluster and I is the index number of clusters those own same hop number. The number of hops is the distance between cluster heads and it is easy to obtain. The super cluster head randomly chooses a cluster head among its one hop neighbor clusters

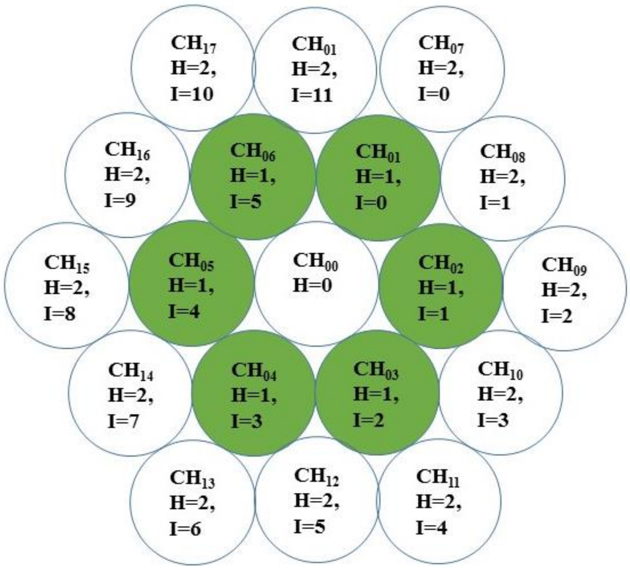


Fig. 5. An example topology

(H, = 1). And assigns index number 0 (I = 0) to it. Starting from chosen cluster index numbers (I = 1, 2, 3, 4, …) are assigned to neighbor clusters turn by turn. The cluster with the number of hops 1 and index number is 0 randomly chooses one of the clusters among its neighbor clusters with the number of hops 2 and assigns index number 0 to it. Clusters with same hop number are assigned index numbers turn by turn. The steps explained above are repeated until all clusters assigned index numbers.

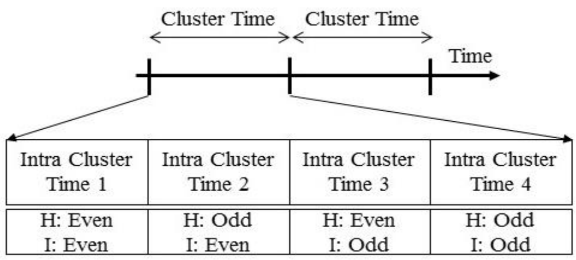


Fig. 6. Time slot assignment

In Fig. 5, the super cluster CH00 chooses CH01 from the clusters with the number of hops 1 and assigns index number 0. CH01 assigns index number 2 to CH02 which is the neighbor cluster with same hop number. In turn, CH02 assigns index number 3 to the CH03. All clusters with the number of hops 1 are assigned index number in the same way. In the same time, CH01 chooses the cluster CH07 from its neighbor clusters with the number of hops 2 and assigns index number 0. To the clusters with the number of hops 2 index numbers are assigned turn by turn.

Obtaining intra cluster time: After all clusters obtain their own relative position information each cluster calculates its time slot which we call intra cluster time. As mentioned above time channel is divided into cluster time and each cluster time consists of 4 intra cluster time which is shown in Fig. 6. Intra cluster time 1 is used by clusters with an even number of hops and an even index number.

The clusters with an odd number of hops and an even index number use intra cluster time 2. Intra cluster time 3 is utilized by clusters with an even number of hops and an odd index number. And intra cluster time 4 is used by clusters with an odd number of hops and an odd index number. Different from normal clusters super cluster uses both intra cluster time 1, 3.

III. PERFORMANCE ANALYSIS

In this section, we compare COD-TS and proposed protocol through simulation using network simulator ns-3. In the network, 20 clusters of nodes are distributed randomly unless specified otherwise. Each cluster consists of 8 nodes including a cluster head. All nodes' modems are configured to 1100 m transmission range and 667 bps transmission rate, to each packet, a preamble is added with duration of 1.5s by the physical layer and propagation speed of acoustic carrier is set to 1500 m/s. Transmission power is set to 2 W, and power consumption of node in reception and idle states are both set to 158 m W. For COD-TS, the transmission power for the schedule packet is set to 5 W. All nodes, including cluster heads, generate packets of size 400 bytes following a Poisson process with mean λ rate. Each node is allowed to send at most 4 frames in a round. Queue size of each node is set as much as big enough to prevent overflow of frames. Simulation run time is 10000 seconds.

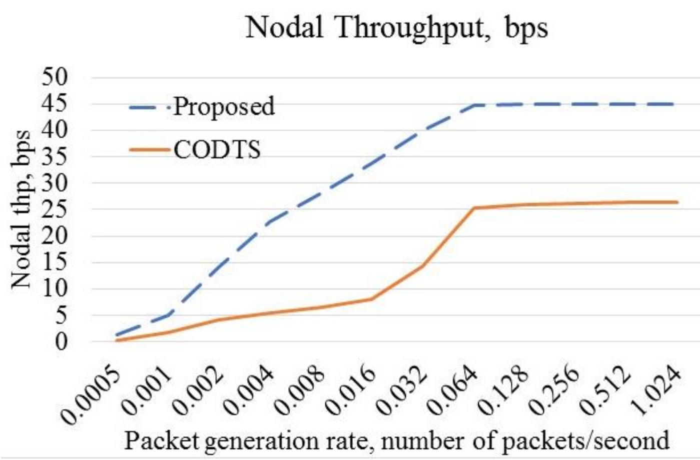


Fig. 7. Nodal throughput with varying packet generation rate λ

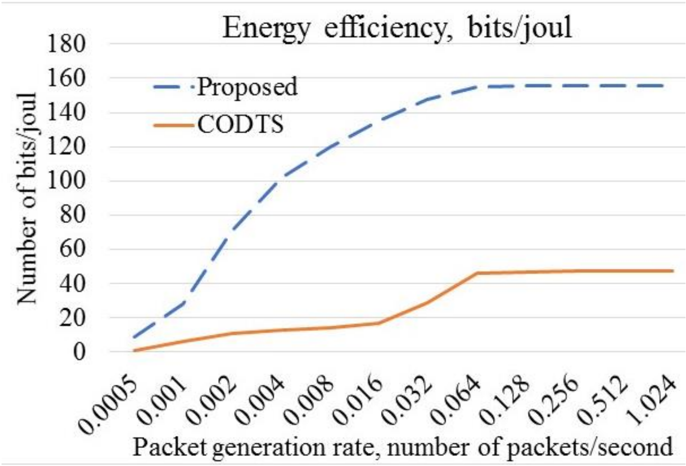


Fig. 8. Energy efficiency with varying packet generation rate λ

For COD-TS, we set D max maximal communication round time to 760 seconds. And for proposed protocol T_{our} duration of an intra cluster time is set to 35 seconds, which is obtained best value by running simulations. Simulation results are averaged from simulation runs with 20 randomly deployed topologies with 20 times repetition for each topology. We measured nodal throughput and energy efficiency with the varying packet generation rate, packet size and number of nodes in the network. Nodal throughput is an average number of successfully received bits in a second per node during simulation time:

Energy Efficiency = Number of Successful Packets * Packet Size / Total Utilized Energy

We discuss simulation results in the following order: first, with varying packet generation rate λ, then, with varying packet size, and then, with varying number of nodes in the network. Fig. 7, 8 show nodal throughput and energy efficiency with varying λ rate. As shown in Fig. 7, proposed protocol achieves higher nodal throughput than COD-TS. COD-TS achieves very low nodal throughput at low packet generation rate. This is because COD-TS spends more time for sending schedule packets more frequently compared to useful data packet transmissions. Also, the schedule packet itself collides with transmissions of nodes of neighboring clusters. On the contrary, the nodal throughput of proposed protocol linearly increases at low packet generation rates and it switches to saturation state at high packet generation rates. Proposed protocol achieves almost three times more energy efficient than COD-TS as it is shown in Fig. 8. The reason for this is similar to the reason explained for nodal throughput. COD-TS transmits schedule packets at a higher power level, as mentioned above the schedule packet itself cause collisions to nodes of neighboring clusters, and because of incorrectly received schedule packets data

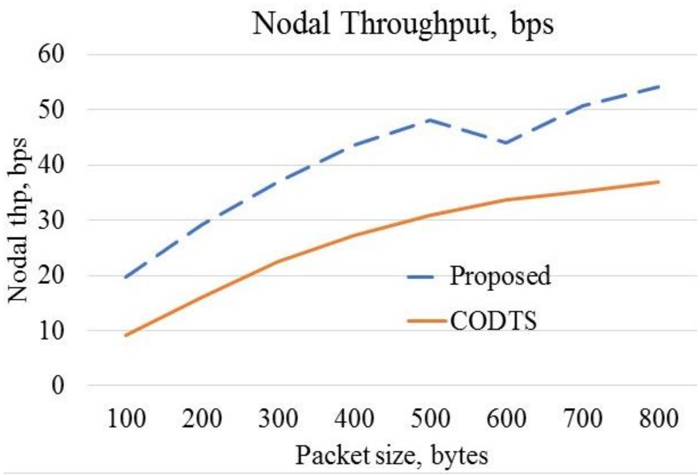


Fig. 9. Nodal throughput with varying packet size

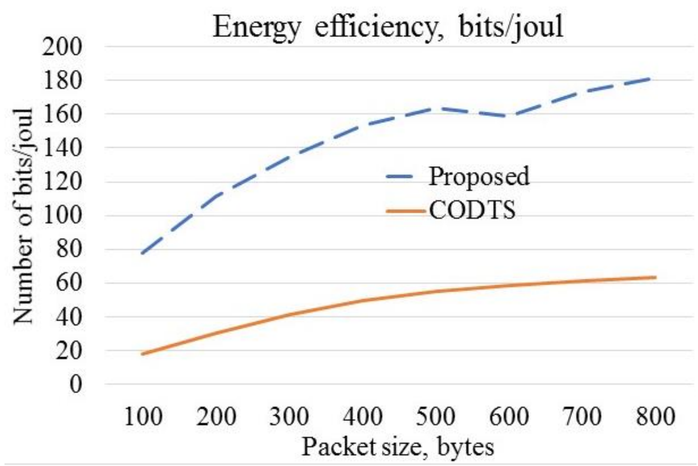


Fig. 10. Energy efficiency with varying packet size

data packets collide transmitted by nodes of neighboring clusters, those all decrease energy efficiency. Performance comparisons of the protocols in terms of nodal throughput and energy efficiency with varying packet size are shown in Fig. 9, 10. Both protocols throughput and energy efficiency increase with increasing packet size. Since the proportion of control packet (overhead) size on useful data size decreases when the payload of data packet size increases. Therefore, throughput that is an average number of useful data bits per second and energy efficiency that is an average number of useful data bits per utilized joule of energy increases when increases data packet size. Proposed protocol outperforms COD-TS. In Fig. 11, 12, comparisons of nodal throughput and energy efficiency with varying number (i.e. density) of nodes in the network are illustrated. As expected, in general for both protocols the performance goes down with increasing density of nodes in the network. For all density of the network, the proposed protocol achieves higher performance with significant difference than COT-TS. Especially, at low density proposed protocol achieves more than two times higher throughput and more than three times higher energy efficiency than that COD-TS achieves. In COD-TS, transmitting same schedule packet to less number of neighbor clusters becomes more overhead than that transmitting it to more number of neighbor clusters. The transmission time of schedule packet does not depend on the number of receivers.

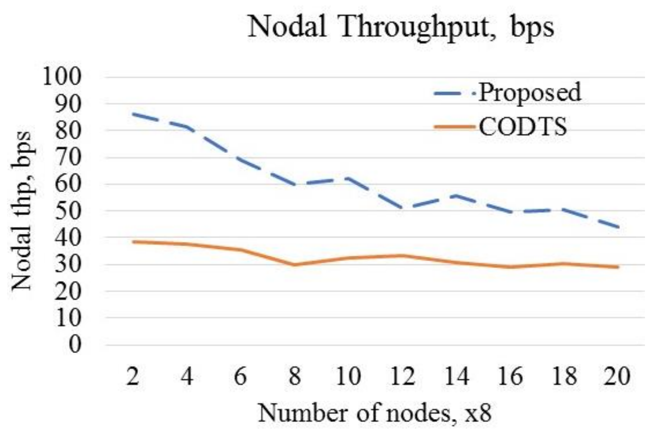


Fig. 11. Nodal throughput with varying number of nodes in the network

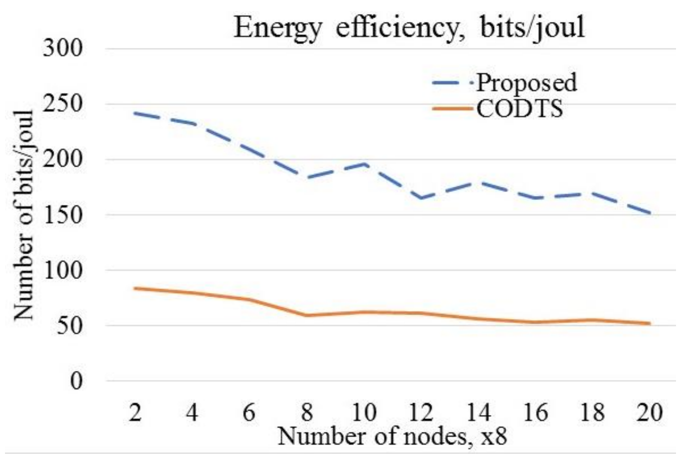


Fig. 12. Energy efficiency with varying number of nodes in the network

Both, throughput and energy efficiency of COD-TS fluctuates as density increases. This is explained as an average number of neighbor clusters is higher with certain numbers of clusters in the network. For example, when number of nodes in the network is 8x8 (8 clusters with 8 nodes each) the throughput goes up since the average number of neighbor clusters of a cluster is higher compared to that when the number of nodes in the network is 6x8. Thus, transmission of schedule packet in the network with 8 clusters is more effective than in the network with 6 clusters.

IV. CONCLUSIONS

The CTSA is a distributed protocol which can work in the network with randomly located nodes. Nodes are clustered, and each cluster calculates its intra cluster time according to its relative position to its neighbors. Relative position information of a cluster includes number of hops from the super cluster head and index number (H, I). In each cluster, a cluster head schedules its members' transmission for the duration of the cluster's intra cluster time. Simulation results show that proposed protocol achieves higher performance with significant difference comparing to COD-TS in terms of nodal throughput and energy efficiency.

ACKNOWLEDGEMENTS

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2015R1D1A1A01059644).

REFERENCES

[1] Keyu Chen, Maode Ma, En Cheng, Fei Yuan, and Wei Su, "A Survey on MAC Protocols for Underwater Wireless Sensor Networks," *IEEE Communications surveys & tutorials,* vol. 16, no. 3, pp. 1433-1447, 2014.

[2] Y. Zhu, Z. Peng, J.-H. Cui, and H. Chen, "Toward Practical MAC Design for Underwater Acoustic Networks," *IEEE Transactions on Mobile Computing*, vol. 4, no. 4, pp. 872-886, 2015.

[3] Yeo, Jin-Ki, Young-Kon Lim, and Heung-Ho Lee. "Modified mac (media access control) protocol design for the acoustic-based underwater digital data communication." *Industrial Electronics, 2001. Proceedings. ISIE 2001. IEEE International Symposium on*. Vol. 1. IEEE, 2001.

[4] Pompili, Dario, Tommaso Melodia, and Ian F. Akyildiz. "A CDMA-based medium access control for underwater acoustic sensor networks." *IEEE Transactions on Wireless Communications* 8.4 (2009): 1899-1909.

[5] Basagni, Stefano, et al. "Optimizing network performance through packet fragmentation in multi-hop underwater communications." *OCEANS 2010 IEEE-Sydney*. IEEE, 2010.

[6] Basagni, Stefano, et al. "Optimized packet size selection in underwater wireless sensor network communications." *IEEE Journal of Oceanic Engineering* 37.3 (2012): 321-337.

[7] Jiarong Zhang, Gang Qiao, and Can Wang, "A Distance Aware Protocol for Underwater Acoustic Communication Networks", *IEEE Transactions on Information and Communications* Technology (2013):1- 6.

[8] Pescosolido, Loreto, Chiara Petrioli, and Luigi Picari. "A multi-band Noise-aware MAC protocol for underwater acoustic sensor networks." *WiMob*. 2013.

[9] Zhang, Jiarong, Gang Qiao, and Feng Zhou. "A relative clock based and energy sense MAC protocol for underwater acoustic communication networks." *OCEANS 2014-TAIPEI*. IEEE, 2014.

[10] J. Liu, Z. Zhou, Z. Peng, and J.-H. Cui, "Mobi-sync: Efficient time synchronization for mobile underwater sensor networks," *in Proceedings of IEEE GLOBECOM, 2010, pp. 1–5.*

[11] F. Lu, D. Mirza, and C. Schurgers, "D-sync: Doppler-based time synchronization for mobile underwater sensor networks," *in Proceedings of ACM Fifth ACM Int. Workshop UnderWater Net., 2010,* pp. 3:1-3:8.

[12] S. Bandyopadhyay and E. J. Coyle, "An energy efficient hierarchical clustering algorithm for wireless sensor networks," in Proc. IEEE INFOCOM, 2003, pp. 1713–1723.

Muzaffar Ergashev, and Sunmyeng Kim. "Clustering Based Time Slot Assignment Protocol for Improving Performance in Underwater Acoustic Networks." Invention Journal of Research Technology in Engineering & Management (IJRTEM), vol. 2, no. 5, 22 May 2018, pp. 66–73., www.ijrtem.com